MMAT5390: Mathematical Image Processing

Assignment 4

Due: 23:59 Wednesday, April 2, 2025

Please give reasons in your solutions unless otherwise specified.

1. Consider a digital image processing scenario where you are working with a high-resolution grayscale image I of size 2048×2048 pixels. The image is represented as $I = (I(m,n))$, where $-1024 \leq m, n \leq 1023$. You are tasked with applying a Butterworth high-pass filter H to the image in the frequency domain to enhance its edges and fine details.

   The Butterworth high-pass filter is defined by its cutoff frequency D_0 and order n. The filter is applied to the Discrete Fourier Transform (DFT) of the image, denoted as $\text{DFT}(I) = (\hat{I}(u,v))$ where $-1024 \leq u, v \leq 1023$, resulting in the filtered frequency domain representation $G(u,v)$.

   Suppose that after applying the filter, you observe the following relationships for specific frequency components:

   $$G(1,1) = \frac{1}{65}\hat{I}(1,1) \quad \text{and} \quad G(2,2) = \frac{1}{17}\hat{I}(2,2),$$

   provided that $\hat{I}(1,1) \neq 0$ and $\hat{I}(2,2) \neq 0$.

   (a) Determine the cutoff frequency D_0 and the order n of the Butterworth high-pass filter based on the given relationships.

   (b) After determining D_0 and n, explain how the filter would affect the image if the order n were increased.

2. Consider a scenario in image processing where you are given a blurred image g of size $N \times N$. The image g is obtained by averaging a sequence of μ consecutive frames from a static scene. The blurred image g is represented as:

   $$g(m,n) = \frac{1}{\mu}\sum_{k=0}^{\mu-1} f(m, n-k) \quad \text{for } 0 \leq m, n \leq N-1,$$

   where:

   - f is the original (unblurred) image, periodically extended to handle boundary conditions.
   - μ is a positive integer in the range $[1, N]$, representing the number of frames averaged.

   Your task is to analyze the degradation process in the frequency domain and derive the degradation function $H(u,v)$ that relates the Discrete Fourier Transform (DFT) of the blurred image g to the DFT of the original image f.

   (a) Show that the DFT of the blurred image g, denoted as $\text{DFT}(g)(u,v)$, can be expressed as:

   $$\text{DFT}(g)(u,v) = H(u,v) \cdot \text{DFT}(f)(u,v) \quad \text{for all } 0 \leq u, v \leq N-1,$$

   where $H(u,v)$ is the degradation function in the frequency domain given by:

   $$H(u,v) = \begin{cases} \frac{1}{\mu}\frac{\sin\left(\frac{\mu\pi v}{N}\right)}{\sin\left(\frac{\pi v}{N}\right)} e^{-\pi j \frac{(\mu-1)v}{N}} & \text{if } v \neq 0, \\ 1 & \text{if } v = 0. \end{cases}$$

   (b) Discuss how the parameter μ affects the shape and behavior of $H(u,v)$.

3. Consider a 4×4 periodically extended image $I = (I(k,l))_{0 \le k,l \le 3}$ given by:

$$I = \begin{pmatrix} a & b & b & b \\ b & a & b & b \\ b & b & a & b \\ b & b & b & a \end{pmatrix},$$

where a and b are distinct positive numbers.
The Gaussian low-pass filter H of variance σ^2 is defined by:

$$H(u,v) = exp(-\frac{u^2+v^2}{\sigma^2}).$$

Let $I_1(u,v) = H_1(u,v)DFT(I)(u,v)$, where H_1 is the Gaussian low-pass filter of variance ab.
Let I_{Haar} be the Haar transform of I.

$$I_{Haar} = \tilde{H} I \tilde{H}^T,$$

where $\tilde{H}$ is the Haar Transform matrix for a 4×4 image.
Suppose $I_1(2,2) = e^{-1}DFT(I)(2,2)$ and $I_{Haar}(1,1) = 2$. Find a and b.

4. Let $f \in M_{N\times N}(\mathbb{R})$ be an image. We use central difference to approximate the derivatives in this question. i.e.

$$\frac{\partial f}{\partial x} \approx \frac{1}{2h}(f(x+h,y) - f(x-h,y))$$
$$\frac{\partial f}{\partial y} \approx \frac{1}{2h}(f(x,y+h) - f(x,y-h))$$

   (a) Find the discrete Laplacian operator Δf.
   (b) Prove that $DFT(\Delta f)(u,v) = H(u,v)F(u,v)$ for some $H \in M_{N\times N}(C)$, where$F = DFT(f)$. Find $H(u,v)$ as a trigonometric polynomial in $\frac{\pi u}{N}$ and $\frac{\pi v}{N}$, i.e. as a polynomial in $sin\frac{\pi u}{N}$, $cos\frac{\pi u}{N}$, $sin\frac{\pi v}{N}$ and $cos\frac{\pi u}{N}$.

The following section is a coding exercise
Instruction: For each task, carefully review the provided MATLAB file or Jupyter Notebook file in the attached zip folder. These files contain missing lines of code that you will need to complete. You may choose to work in either MATLAB or Python to address the tasks.

- Add the missing lines of code to complete the functionality as described.
- Test your implementation using the provided grayscale image.

Note: This assignment focuses on grayscale image processing. Ensure your solutions are tailored to handle grayscale images appropriately.

5. **(optional)** In this task, you will reconstruct an image from its frequency domain representation. The frequency domain representation of an image is obtained using the Discrete Fourier Transform (DFT), which can be expressed as a matrix multiplication operation.

   The DFT of an image g can be written as:

$$\hat{g} = UgU^T$$

   where:

   - U is the DFT matrix defined as:

$$U_{\alpha\beta} = \frac{1}{\sqrt{N}} e^{-2\pi j \frac{\alpha\beta}{N}}, \quad 0 \le \alpha, \beta \le N-1$$

   - $\hat{g}$ is the matrix of Fourier coefficients.

   Your task is to reconstruct the original image g from a modified version of $\hat{g}$. You are **not allowed** to use built-in functions like `ifft2` in MATLAB or any Fourier transform modules in Python (e.g., `numpy.fft` or `scipy.fft`). Instead, you must implement the inverse DFT manually using matrix operations.

6. **(optional)** In this task, we use the built-in function ($fft2$($ifft2$) and $fftshift$($ifftshift$) in MATLAB or $numpy.fft$ in Python) to simplify our code. The only difference between $fft2$ (or $numpy.fft.fft2$) and our definition of DFT is that our definition is $\frac{1}{N^2}$ of $fft2$. Similarly, Our definition of iDFT is N^2 of $ifft2$.

   The built-in function $fftshift$ and $ifftshift$ in MATLAB or the corresponding implementations in $numpy.fft$ module have the same effect as $circshift(x, [h/2, w/2])$, which shifts frequency components at the corner to the center. So we use the two functions to replace $circshift$ to simplify the code.

   In order to compute the distance between each element and the center, we make use of the built-in function $meshgrid$ in MATLAB or $numpy.meshgrid$ in Python. Here is an example of the function $[X, Y] = meshgrid([-2:2], [-1:1])$:

$$X = \begin{pmatrix} -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \end{pmatrix} \qquad Y = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

   You are required to define the Gaussian Low Pass Filter using the definition

$$H(u, v) = e^{-\frac{u^2+v^2}{2\sigma^2}}$$

   where $-\frac{N}{2} \leq u \leq \frac{N}{2} - 1$ and $-\frac{N}{2} \leq v \leq \frac{N}{2} - 1$.

   The results are as follows:

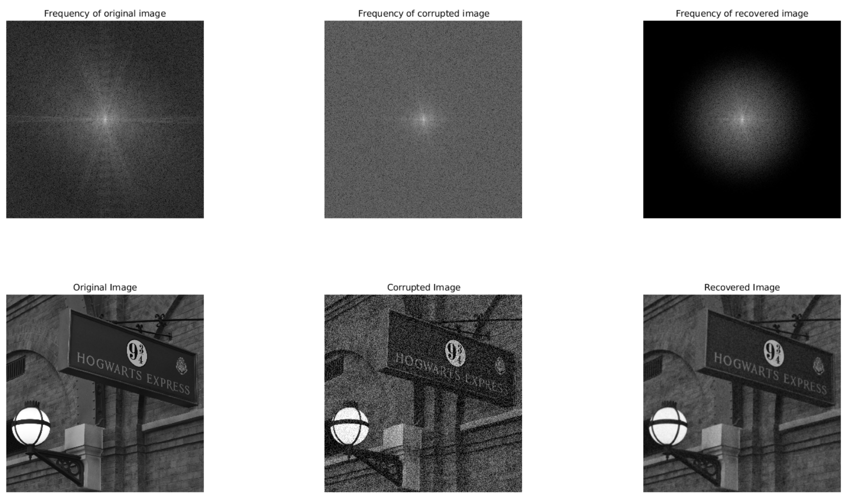

Figure 1: Experimental Results